

Compact Representation of Photosynthesis Dynamics by Rule-based Models (Full Version)

L. Brim, J. Nižnan, D. Šafránek *

Faculty of Informatics, Masaryk University
Brno, Czech Republic
safranek@fi.muni.cz

Traditional mathematical models of photosynthesis are based on mass action kinetics of light reactions. This approach requires the modeller to enumerate all the possible state combinations of the modelled chemical species. This leads to combinatorial explosion in the number of reactions although the structure of the model could be expressed more compactly. We explore the use of rule-based modelling, in particular, a simplified variant of Kappa, to compactly capture and automatically reduce existing mathematical models of photosynthesis. Finally, the reduction procedure is implemented in BioNetGen language and demonstrated on several ODE models of photosynthesis processes. This is an extended version of the paper published in proceedings of 5th International Workshop on Static Analysis and Systems Biology (SASB) 2014.

1 Introduction

Photosynthesis is one of the most important biophysical processes driving life on Earth. Most life forms, including humans, depend on photosynthesis that transforms energy of solar radiation into energy-rich organic matter, releases oxygen that we breathe, and removes excess carbon dioxide from the atmosphere that would threaten the Earth's energy balance. Adding to the relevance of photosynthesis, significant expectations emerged lately in connection with potential human interventions in the global carbon cycle – among the considered alternatives are the higher generation biofuels [1] or biomineralization by point-source carbon capture [10].

Current coarse-grained mathematical models of photosynthesis [12] cover the known parts of the entire process. They build up the light reactions dynamics from simplified interactions on and inbetween complicated protein complexes involved in the transfer of the energy from light into the cell. Many different local modifications at these protein structures are traversed after reception of the photon. To capture the process mechanistically, many elementary chemical reactions connect together to form the model. Each effective structure combination has to be enumerated in order to assign the appropriate kinetic laws. This inevitable expansion then leads to combinatorial explosion in the number of possible complexes.

In [21] we have developed an online repository for mathematical models of photosynthesis. That effort has opened many questions regarding the differing levels of available models and the problem of their formal representation in a single suitably expressive formalism. After several years of interactions with modellers targeting photosynthesis, we now attempt to move towards practical applications of state-of-the-art formal methods in that field.

Rule-based modelling [7, 4, 3] is an approach that has been developed to tackle primarily the complexity of cell signalling systems where combinatorial explosion comes from configurations of phosphate

*The work has been supported by the EC OP project No. CZ.1.07/2.3.00/20.0256.

bindings to specific sites of a protein. In particular, it allows us to compactly represent complicated models that would be tedious to specify using traditional reaction-based methods [5, 11]. The interactions between proteins are represented using rules at the level of functional components. In photosynthesis, there occurs a number of specific protein complex modifications that are in abstract essence similar to phosphorylation though crucially different at the side of physics. The two most well-known representatives of rule-based modelling languages are Kappa [7] and BioNetGen Language [8].

On the theoretical side, we contribute to the increasing set of algebraic-based modelling efforts by employing a simplified version of Kappa calculus to compact a set of domain-specific models coming from biophysics of photosynthesis. We do not construct the models from the scratch but we rather take several existing kinetic models of photosynthesis-related processes and reformulate them in the algebraic framework. At the level of compositional representation, we formulate syntactic reductions of the models that preserve behavioural equivalence.

On the practical side, we employ BioNetGen language (BNGL) and related tools [8, 20] to implement the models. We contribute by automatising our reduction procedure in Python. By applying reductions to the considered models we show the power of process-algebraic framework to compactly represent combinatorially exploding systems of light reactions.

The paper shows the importance of process-algebraic description for the domain of photosynthesis. To the best of our knowledge, the application of such techniques in the field of systems biology of photosynthesis is still at its beginning and we believe our contribution is useful to help to establish rule-based modelling in the domain.

1.1 Related Work

There are many applications of rule-based modelling available (see [3] for an overview). However, applications to photosynthesis are very rare. In [22], the authors provide a model of chlorophyll *a* fluorescence induction kinetics that is simulated in the rule-based framework by means of Monte Carlo simulation. The work is unique in the sense it pioneers rule-based approach for photosynthesis models. The contribution brings a stochastic simulation algorithm that reflects the differing context where the context-free rules appear. This is important, since the quantitative rates of electron transfers are modulated by aggregation of several modifications of the protein complex, i.e., the photosystem. The considered model is comparable to Lazár model [14] we use for our case study.

In [2], the problem of combinatorial complexity in models where the quantitative semantics cannot be generated is addressed. Several frameworks for abstracting the models at the level of semantics have been developed for Kappa.

2 Background

We define simplified Kappa using a process-like notation as is presented in [6], syntax and the notions of structural equivalence and matching are entirely take from [6]:

| | | | |
|------------|---------------------------------------|----------------|---|
| expression | $E ::= \emptyset \mid a, E$ | site | $s ::= n_i^\lambda$ |
| agent | $a ::= N(\sigma)$ | site name | $n ::= x \in \mathcal{S}$ |
| agent name | $N ::= A \in \mathcal{A}$ | internal state | $\iota ::= \varepsilon \mid m \in \mathbb{V}$ |
| interface | $\sigma ::= \emptyset \mid s, \sigma$ | binding state | $\lambda ::= \varepsilon \mid i \in \mathbb{N}$ |

where \mathcal{A} is a finite set of agent names, \mathcal{S} is a finite set of site names, \mathbb{V} is a finite set of values representing modified states of the sites. An agent is denoted by its name and its interface. Interface

consists of a sequence of sites. x_t^λ denotes a site x with internal state t and binding state λ . If the binding state is ε then the site is free, otherwise it is bound. By convention, when a binding or internal site is not specified, ε is considered.

Note that full Kappa is richer. It allows a binding state meaning a *free or bound site*, denoted by a question mark. We also omit rates from the rules.

Definition 2.1 *An expression is well-formed if a site name occurs only once in an interface and if each binding state ($\neq \varepsilon$) present in the expression occurs exactly twice. The set of all well-formed expressions is denoted as \mathcal{E} . The set of all well-formed expressions that can be generated from the literal a is called the set of all well-formed agents and is denoted as \mathcal{E}_a . Similarly, \mathcal{E}_σ denotes the set of all well-formed interfaces and \mathcal{E}_s the set of all well-formed sites.*

Next, we define some notations we use throughout the text.

Notation 2.1 expressions $E, E' \in \mathcal{E}$; agents $a, a' \in \mathcal{E}_a$; agent name $A \in \mathcal{A}$; interfaces $\sigma, \sigma' \in \mathcal{E}_\sigma$; sites $s, s' \in \mathcal{E}_s$; site name $x \in \mathcal{S}$; internal state $t \in \{\varepsilon\} \cup \mathbb{V}$; specific internal state $m \in \mathbb{V}$; binding state $\lambda \in \{\varepsilon\} \cup \mathbb{N}$;

Next, we provide inductive definitions of some useful mappings.

Definition 2.2 Agent name is a mapping $\text{name} : \mathcal{E}_a \rightarrow \mathcal{A}$ defined as $\text{name}(A(\sigma)) = A$.

We define agent sites as a mapping $\text{sites} : \mathcal{E}_a \rightarrow 2^{\mathcal{S}}$ such that $\text{sites}(A()) = \emptyset$, $\text{sites}(A(x_t^\lambda)) = \{x\}$, and $\text{sites}(A(s, \sigma)) = \text{sites}(A(s)) \cup \text{sites}(A(\sigma))$.

Agent internal state is a mapping $\text{state} : \mathcal{E}_a \rightarrow (\mathbb{V} \cup \{\varepsilon\})^{\mathcal{S}}$ defined as $\text{state}(A()) = \emptyset$, $\text{state}(A(x_t^\lambda)) = \{(x, t)\}$, and $\text{state}(A(s, \sigma)) = \text{state}(A(s)) \cup \text{state}(A(\sigma))$.

Definition 2.3 Structural equivalence $\equiv \subseteq \mathcal{E} \times \mathcal{E}$ is defined as a relation satisfying the following properties:

1. Reflexivity: $E \equiv E$
2. The order of sites in interfaces does not matter:

$$E, A(\sigma, s, s', \sigma'), E' \equiv E, A(\sigma, s', s, \sigma'), E'$$
3. The order of agents in an expression does not matter:

$$E, a, a', E' \equiv E, a', a, E'$$
4. Binding states can be injectively renamed:

$$E[i/j] \equiv E$$

where $i, j \in \mathbb{N}$ and i does not occur in E .

Solution $[E] \in 2^{\mathcal{E}}$ denotes the equivalence class of E in \equiv . \mathcal{L} is a set of all solutions.

Definition 2.4 A rule is a pair of expressions E_l, E_r (usually written as $E_l \rightarrow E_r$). The set of all rules is denoted as \mathcal{R} .

The left hand side E_l of the rule describes the solution taking part in the reaction and the right hand side E_r describes the effects of the rule. The rule can be either a binding rule or a modification rule. A binding (unbinding) rule binds two free sites together (or unbinds two bound sites). A modification rule modifies some internal state [6].

Definition 2.5 Matching is a relation denoted as $\models \subseteq \mathcal{E} \times \mathcal{E}$ and defined inductively in the left column below. Replacement is a function $\mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E}$ defined in the right column.

$$\begin{array}{ll}
n_i^\lambda \models n_i^\lambda & n_i^\lambda [n_{i_r}^{\lambda_r}] = n_{i_r}^{\lambda_r} \\
n_i^\lambda \models n^\lambda & n_i^\lambda [n_{i_r}^{\lambda_r}] = n_{i_r}^{\lambda_r} \\
\sigma \models \emptyset & \sigma[\emptyset] = \sigma \\
\frac{s \models s_l \quad \sigma \models \sigma_l}{s, \sigma \models s_l, \sigma_l} & s, \sigma[s_r, \sigma_r] = s[s_r], \sigma[\sigma_r] \\
\frac{\sigma \models \sigma_l}{N(\sigma) \models N(\sigma_l)} & N(\sigma)[N(\sigma_r)] = N(\sigma[\sigma_r]) \\
E \models \emptyset & E[\emptyset] = E \\
\frac{a \models a_l \quad E \models E_l}{a, E \models a_l, E_l} & (a, E)[a_r, E_r] = a[a_r], E[E_r]
\end{array}$$

A replacement can be applied only if the corresponding matching is satisfied.

In order to apply a rule $E_l \rightarrow E_r$ to a solution $[E]$ the expression E representing the solution must first be reordered to an equivalent expression E' that matches E_l (according to the definition of matching stated above). E' is then replaced with $E'[E_r]$ (also defined above).

Definition 2.6 Rule application is a mapping $\tau : \mathcal{L} \times \mathcal{R} \rightarrow \mathcal{L}$ such that $\tau([E], (E_l, E_r)) = [E'[E_r]]$ whenever $\exists E' \in [E]. E' \models E_l$.

Rules yield a *transition system* between solutions containing an edge $[E] \rightarrow_{E_l, E_r} [E'[E_r]]$ whenever $\exists E' \in [E]. E' \models E_l$.

Definition 2.7 An agent signature (Σ, I) is a pair of mappings $\Sigma : \mathcal{A} \rightarrow 2^{\mathcal{S}}$ and $I : \mathcal{A} \times \mathcal{S} \rightarrow 2^{\mathbb{V}}$.

Informally, Σ restricts for each agent name $A \in \mathcal{A}$ the set of site names that can occur in an agent with name A . And I restricts the set of internal states a particular site can attain.

Definition 2.8 E satisfies agent signature (Σ, I) , denoted $(\Sigma, I) \vdash E$, if E satisfies one of the following conditions:

1. $E \equiv \emptyset$
 2. $E \equiv A()$ and $A \in \text{dom}(\Sigma)$
 3. $E \equiv A(x_\varepsilon^\lambda)$ and $x \in \Sigma(A)$
 4. $E \equiv A(x_m^\lambda)$ and $x \in \Sigma(A)$ and $m \in I(A, x)$
 5. $E \equiv A(s, \sigma)$ where $(\Sigma, I) \vdash A(s)$ and $(\Sigma, I) \vdash A(\sigma)$
 6. $E \equiv E_l, A(\sigma)$ where $(\Sigma, I) \vdash E_l$ and $(\Sigma, I) \vdash A(\sigma)$
- If $r = (E_l, E_r) \in \mathcal{R}$ and $(\Sigma, I) \vdash E_l$ and $(\Sigma, I) \vdash E_r$ then $(\Sigma, I) \vdash r$.
If $R \subseteq \mathcal{R}$ and $\forall r \in R. (\Sigma, I) \vdash r$ then $(\Sigma, I) \vdash R$.

Definition 2.9 An agent a is complete with respect to signature (Σ, I) , denoted $(\Sigma, I) \models a$, if $\text{sites}(a) = \Sigma(\text{name}(a)) \wedge \forall x \in \text{sites}(a). \text{state}(a)(x) \in I(\text{name}(a), x)$.

An expression E is complete with respect to signature (Σ, I) , denoted $(\Sigma, I) \models E$, if it satisfies one of the following conditions:

1. $E \equiv \emptyset$
2. $E \equiv a, E'$ where $a \in \mathcal{E}_a, E' \in \mathcal{E}$ and $(\Sigma, I) \models a$ and $(\Sigma, I) \models E'$

$\mathcal{E}_{(\Sigma, I)} = \{E \in \mathcal{E} \mid (\Sigma, I) \models E\}$ is a set of all expressions that are complete with respect to signature (Σ, I) .

Definition 2.10 A rule-based model \mathcal{M} is a tuple (Σ, I, R) that satisfies the condition $(\Sigma, I) \vdash R$. We use the notation $\text{Signature}(\mathcal{M}) = (\Sigma, I)$, $\text{Rules}(\mathcal{M}) = R$, $\mathcal{M} \vdash E \iff (\Sigma, I) \vdash E$ for $E \in \mathcal{E}$, $\mathcal{M} \models E \iff (\Sigma, I) \models E$ for $E \in \mathcal{E}$, and $\mathcal{E}_{\mathcal{M}} = \mathcal{E}_{(\Sigma, I)}$.

Definition 2.11 An initialised model M is a pair (\mathcal{M}, E_i) where \mathcal{M} is a rule-based model and E_i is an expression representing the initial solution such that $\mathcal{M} \models E_i$.

Definition 2.12 A state space of an initialised model $M = (\mathcal{M}, E_i)$ is a pair $(\text{Solutions}(M) \subseteq \mathcal{L}, \text{Reactions}(M) \subseteq \mathcal{L} \times \mathcal{L})$ defined inductively as follows:

1. $[E_i] \in \text{Solutions}(M)$
2. $[E] \in \text{Solutions}(M)$ and $\exists r \in \text{Rules}(\mathcal{M}). \tau([E], r) = [E']$
if and only if $[E'] \in \text{Solutions}(M)$ and $([E], [E']) \in \text{Reactions}(M)$

Definition 2.13 Initialised models $M_1 = (\mathcal{M}_1, E_1)$ and $M_2 = (\mathcal{M}_2, E_2)$ are structurally equivalent, denoted $M_1 \equiv M_2$, if and only if $\text{Solutions}(M_1) = \text{Solutions}(M_2)$ and $\text{Reactions}(M_1) = \text{Reactions}(M_2)$.

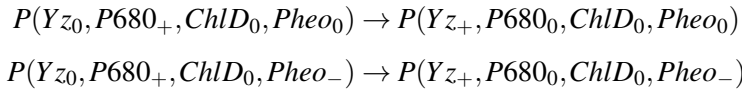
Definition 2.14 Models \mathcal{M}_1 and \mathcal{M}_2 are structurally equivalent, denoted $\mathcal{M}_1 \equiv \mathcal{M}_2$, if and only if $\forall E_i \in \mathcal{E}_{\mathcal{M}_1} \cup \mathcal{E}_{\mathcal{M}_2}. (\mathcal{M}_1, E_i) \equiv (\mathcal{M}_2, E_i)$.

In BNGL, agents are called molecules and they are specified in a similar manner as in the simplified Kappa. An example of a molecule is $A(x \sim n! 1)$ where the site x has an internal state n (separated from the site by a tilde) and a binding state is 1 (separated by the exclamation mark). The BNGL alternatives to agent signatures are called molecule types they are defined using the notation demonstrated in the following example: $A(x \sim n \sim b, y \sim n \sim a)$. Here, the allowed internal states of the individual sites are separated by tildes (site x can have an internal state n or b). Rules are described by the *lhs* \rightarrow *rhs* notation (or *lhs* \leftrightarrow *rhs* in the case of reversible rules). The individual model components (molecule types, reaction rules, seed species, observables) are in BNGL separated by the *begin keyword* and *end keyword* pairs.

3 Model Reductions

In this section, we formally define several syntactic operations that can be used to reduce rule-based models. In particular, we assume an original ODE model to be directly transferred to a rule-based model. At that level we apply syntactic reductions to remove redundancies of the original model. As a motivation the following example can be considered.

Example 3.1 Assume a comprehensive model of photosystem II protein complex containing the following rules:



It is obvious that Pheo is not affected by these rules in any way. Since it has only two possible states (0 and +), the two rules can be reduced into the following single rule where Pheo has an empty internal state:

$$P(Y_{z0}, P680_+, ChlD_0, Pheo) \rightarrow P(Y_{z+}, P680_0, ChlD_0, Pheo)$$

To capture the syntactic manipulation mentioned above, we introduce an operation called *context enumeration elimination* of Pheo in P . Note that it looks like we could eliminate Pheo totally, but notice that it is unbound (i.e. its binding state is ϵ).

Definition 3.1 Model \mathcal{M}_1 and model \mathcal{M}_2 are in relation context enumeration elimination, $((\mathcal{M}_1, \mathcal{M}_2) \in \rho_{cee})$, iff $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2) = (\Sigma, I)$ and $\exists A \in \mathcal{A}, x \in \mathcal{S}, \lambda \in \{\epsilon\} \cup \mathbb{N}, E_l, E_r \in \mathcal{E}, \sigma_l, \sigma_r \in \mathcal{E}_\sigma$ such that

1. $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{(E_l^m, E_r^m) | m \in I(A, x)\}$ where $\forall m \in I(A, x) : E_l^m \equiv E_l, A(\sigma_l, x_m^\lambda)$ and $E_r^m \equiv E_r, A(\sigma_r, x_m^\lambda)$.
2. $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \{(E_l', E_r')\}$ where $E_l' \equiv E_l, A(\sigma_l, x_\epsilon^\lambda)$ and $E_r' \equiv E_r, A(\sigma_r, x_\epsilon^\lambda)$.

Theorem 3.1 Context enumeration elimination preserves structural equivalence of models. If $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{cee}$ then $\mathcal{M}_1 \equiv \mathcal{M}_2$.

Proof Let $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{cee}$. Then $\mathcal{E}_{\mathcal{M}_1} = \mathcal{E}_{\mathcal{M}_2}$. Let $E_i \in \mathcal{E}_{\mathcal{M}_1}$, $M_1 = (\mathcal{M}_1, E_i)$ and $M_2 = (\mathcal{M}_2, E_i)$. We prove that $M_1 \equiv M_2$ by induction through the structure of their state spaces. Without a loss of generality we can fix the variables used in Definition 3.1.

1. From Definition 2.12: $[E_i] \in \text{Solutions}(M_1)$, $[E_i] \in \text{Solutions}(M_2)$
2. *Completeness*:
Let $[E] \in \text{Solutions}(M_1)$ and $r \in \text{Rules}(\mathcal{M}_1) \cdot \tau([E], r) = [E']$.
From induction we have $[E] \in \text{Solutions}(M_2)$.
(a) $r \in \text{Rules}(\mathcal{M}_1) \cap \text{Rules}(\mathcal{M}_2)$. Then we have $r \in \text{Rules}(\mathcal{M}_2)$.
(b) $r \in \text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2)$. So $\exists m \in I(A, x) \cdot r = (E_l^m, E_r^m)$.
Let $r' = (E_l', E_r') \in \text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1)$ and $e \in [E] \cdot e \models E_l^m$. Then from Definition 2.5 we have $e \models E_l'$ and $e[E_r^m] = e[E_r']$. Therefore, $\tau([E], r') = [E']$.

Thus $\text{Solutions}(M_1) \subseteq \text{Solutions}(M_2)$, $\text{Reactions}(M_1) \subseteq \text{Reactions}(M_2)$.

3. *Soundness*:
Let $[E] \in \text{Solutions}(M_2)$ and $r \in \text{Rules}(\mathcal{M}_2) \cdot \tau([E], r) = [E']$.
From induction we have $[E] \in \text{Solutions}(M_1)$.
(a) $r \in \text{Rules}(\mathcal{M}_2) \cap \text{Rules}(\mathcal{M}_1)$. Then we have $r \in \text{Rules}(\mathcal{M}_1)$.
(b) $r \in \text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1)$. So $r = (E_l', E_r')$. Let $e \in [E] \cdot e \models E_l'$. Then there must be x_m^λ in e that gets matched to the x_ϵ^λ part E_l' . It must be that $m \in I(A, x)$ and so $r' = (E_l^m, E_r^m) \in \text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2)$. From Definition 2.5 we have $e \models E_l^m$ and $e[E_r^m] = e[E_r']$. Therefore, $\tau([E], r') = [E']$.

Thus $\text{Solutions}(M_2) \subseteq \text{Solutions}(M_1)$, $\text{Reactions}(M_2) \subseteq \text{Reactions}(M_1)$.

Example 3.2 In Example 3.1 we did not remove Pheo from

$$P(Y_{z0}, P680_+, ChlD_0, Pheo) \rightarrow P(Y_{z+}, P680_0, ChlD_0, Pheo)$$

because it would make this rule applicable in solutions where it was not applicable before the change. The new rule

$$P(Y_{z0}, P680_+, ChlD_0) \rightarrow P(Y_{z+}, P680_0, ChlD_0)$$

could be applied in some solutions where Pheo is bound. Therefore, this operation does not preserve the structural semantics of every model. However, if in an initialized model, Pheo is not initialized in a bound state and there are no rules that would change it into a bound state then the result of this operation is equivalent with the original initialized model.

Definition 3.2 Model \mathcal{M}_1 is in relation generic unbound context elimination with model \mathcal{M}_2 , $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{guce}$, iff $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2)$ and $\exists A \in \mathcal{A}, x \in \mathcal{S}, E_l, E_r \in \mathcal{E}, \sigma_l, \sigma_r \in \mathcal{E}_\sigma$ such that

1. $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{(E_l^1, E_r^1)\}$ where $E_l^1 \equiv E_l, A(\sigma_l, x_\varepsilon^e)$ and $E_r^1 \equiv E_r, A(\sigma_r, x_\varepsilon^e)$,
2. $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \{(E_l^2, E_r^2)\}$ where $E_l^2 \equiv E_l, A(\sigma_l)$ and $E_r^2 \equiv E_r, A(\sigma_r)$.

This reduction is useful in models where the removed contexts are guaranteed not to be bound.

Example 3.3 Models of photosynthesis can contain rules such as the following antenna deexcitation rule:

$$P(P680_n, Qa_n, PhD_n, ac_*, ChlD_n) \rightarrow P(P680_n, Qa_n, PhD_n, ac_n, ChlD_n)$$

We can see that this rule contains a lot of specific contexts – sites with specified internal states whose internal state is not changed in the rule $(P680_n, Qa_n, PhD_n, ChlD_n)$. We want to remove these contexts if they do not change the semantics of the model. We can define an operation that eliminates a single context from a rule. After the operation is applied to an initialized model we can check if the original and the resulting models have the same state space.

Motivated by the previous example, we formally define the operation of removing a specific context from a model.

Definition 3.3 Model \mathcal{M}_1 is in relation specific unbound context elimination with model \mathcal{M}_2 , $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{suce}$, iff $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2)$ and $\exists A \in \mathcal{A}, x \in \mathcal{S}, E_l, E_r \in \mathcal{E}, \sigma_l, \sigma_r \in \mathcal{E}_\sigma, m \in \mathbb{V}$ such that

1. $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{(E_l^1, E_r^1)\}$ where $E_l^1 \equiv E_l, A(\sigma_l, x_m^e)$ and $E_r^1 \equiv E_r, A(\sigma_r, x_m^e)$,
2. $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \{(E_l^2, E_r^2)\}$ where $E_l^2 \equiv E_l, A(\sigma_l)$ and $E_r^2 \equiv E_r, A(\sigma_r)$.

This reduction can be used if the set of reachable solutions in an intialized model to which this rule can be applied is not affected by the reduction.

Sometimes it may be useful to eliminate a rule from a model. Reasons for doing so can be different. For example, one might want to see how the behaviour of a model changes after the rule is removed. Or if the rule is not reachable in an initialised model then it can be removed to reduce the size of the model description.

Definition 3.4 Model \mathcal{M}_1 is in relation rule elimination with model \mathcal{M}_2 , denoted $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{re}$, if and only if $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2)$ and $\exists r \in \mathcal{R}$ such that $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{r\}$ and $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \emptyset$.

If a rule is not reachable in some initialised model then we can safely remove it without affecting the semantics of the initialised model.

4 Application to Photosynthesis Models

In this section, we describe the application of the reductions to several models of photosynthesis. If the model is reaction-based, we first rewrite it to rule-based form.

4.1 Implementation

We used the library PySB [15] for the specification of the photosynthetic models and to automatise their export to BNGL. The scripts that implement syntactic operations have been written in Python. We used BioNetGen for constructing and simulating the models. The scripts are available at <https://github.com/jniznan/rbm-photosynthesis>.

We search the space of possible models that can be constructed by applying syntactic operations to the original model by depth-first search. We stop when we find a model that cannot be further reduced. There can be multiple models that cannot be further reduced. Our algorithm finds only one. We apply the syntactic operations in a given order: (i) context enumeration elimination, (ii) generic/specific context elimination, (iii) rule elimination. This approach is a heuristic that attempts to maximize the number of reductions.

4.2 Photosynthesis

Light-dependent reactions begin in photosystem II where the photons hit and excite the antenna molecules. The excitation then travels via a chain of proteins until it arrives to chlorophyll *a*. Or a photon can directly excite chlorophyll *a*. This excitation causes the primary electron acceptor (pheophytin) to accept an electron from chlorophyll *a* species called P680 – the primary electron donor. The electron is exchanged by multiple protein molecules until it reaches plastoquinone. The electron missing from chlorophyll *a* is replenished through a tyrosine residue from so called oxygen-evolving complex that strips electrons from water molecules, producing molecular oxygen and hydrogen protons into the lumen.

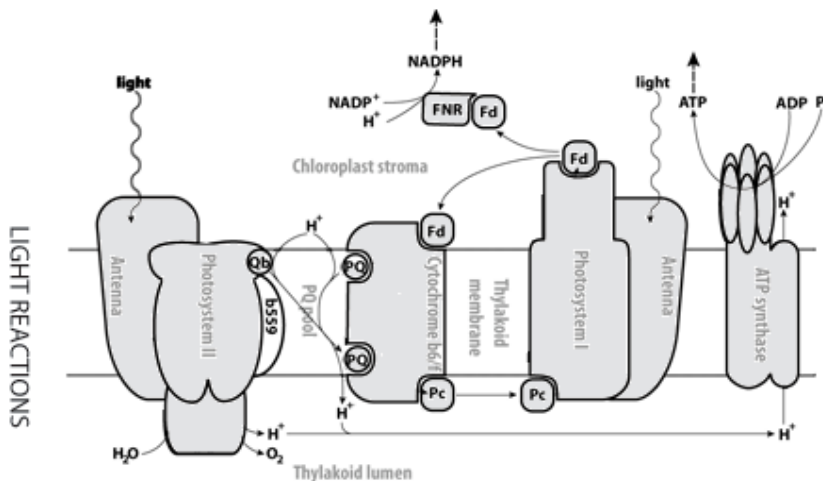


Figure 1: Schema of the thylakoid membrane where light-dependent reactions occur.

After plastoquinone accepts two electrons, it is converted to its PQH_2 form by accepting two hydrogen protons from the chloroplast stroma. Then it travels to cytochrome b_6f where it is converted back to

its original PQ form, leaving the two electrons in the cytochrome and the two hydrogen protons in the lumen. After that, plastoquinone returns to photosystem II, ready to accept other electrons. As is shown in Figure 1, the electrons travel through plastocyanin, photosystem I, ferredoxin to ferredoxin-NADP reductase where they are used for converting NADP^+ to NADPH. The process of the electrons travelling from the oxygen-evolving complex to the ferredoxin-NADP reductase is known as the *Z-scheme* of light. Hydrogen protons that are left in the lumen are pumped back into the chloroplast stroma by powering ATP synthase which uses that energy to convert ADP into ATP.

4.3 Comprehensive Model of Photosystem II

Photosynthesis is well-adapted for spatially and temporally heterogeneous environments [16]. The light conditions under which photosynthesis occurs are constantly changing for many reasons, including day-night cycle, moving clouds or changing seasons. Mechanisms lying behind the adaptability of photosynthesis are not yet fully understood [19]. Evidence confirms that photosynthesis in fluctuating light is more dynamic than simply adapting to the light extremes.

Rules are used to informally capture the model, but no existing formal rule-based language is employed. This fact makes the CMS a perfect candidate for rewriting it into an RBM format. Since the reaction rates are not available the model is worked out at a qualitative level of view.

The original model contains 22 rules. By applying several syntactic operations we are able to reduce the model size to just 17 rules with smaller contexts.

We have also considered a reduced variant of the model that concentrates on light absorption by electron transfers inside PSII. That allowed the authors [9] to introduce a fully specified kinetic model. By employing the reductions we were able to compact the model rules but not to decrease their number.

4.3.1 Reduction Procedure

In an idealized case, PSII consists of 9 principal components with several possible states: Peripheral antenna (ground, excited), Core antenna (ground, excited), Mn Cluster (S_0, S_1, S_2, S_3, S_4), Y_z (neutral, oxidized), P680 (neutral, oxidized), ChlD (ground, excited, oxidized), Pheo (neutral, reduced), Q_A (neutral, reduced), and Q_B (neutral, reduced, 2-reduced, PQH_2 , empty).

When modelling PSII using chemical reactions or ODEs, each state in the model is a combination of the states of its components. This leads to 4800 possible combinatorial states. In [16], the authors argue that this combinatorial complexity makes a highly comprehensive model, that uses all these components and states, practically impossible. Such model could contain $4800 \times 4800 = 23.04 \times 10^6$ reactions. This complexity would get larger if this comprehensive model were to include other models of similar complexity. Instead of trying to build this large model, the authors [16] reduce the system to model the behaviour they are interested in.

The system is reduced to model the effect of exposing PSII to ultra-short saturating flash of light on the fluorescence emission [16]. The model that the authors reduce is based on [18] and [9]. They assume that the fluorescence detectors used in measurements are not fast enough to capture processes occurring during the ultra-short flash of light. Therefore, they are explicitly using an initial state of the model with antenna already excited. Due to the choice of the modelled behaviour they do not consider the peripheral components of PSII (the Mn cluster, Y_z and Q_B). Another type of reduction they demonstrate is the biochemical removal of the peripheral antenna, leaving only core antenna and chlorophyll donor to hold the excitation.

The reduced model contains only the following 5 components leading to 48 possible combinatorial states: Core antenna (ground, excited), P680 (neutral, oxidized), ChlD (ground, excited, oxidized), Pheo (neutral, reduced), and Q_A (neutral, reduced). The following is the rule-based BNGL representation of the model:

```
begin molecule types
  PSII(P680~n~p,Qa~n~m,PhD1~n~m,ac~n~exc,ChlD1~n~p~exc)
end molecule types

begin reaction rules
  #deexcitation_antenna_fluorescence:
  PSII(P680~n,Qa~n,PhD1~n,ac~exc,ChlD1~n) -> PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~n)
  #deexcitation_antenna_heat:
  PSII(P680~n,Qa~n,PhD1~n,ac~exc,ChlD1~n) -> PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~n)
  #deexcitation_Chld1_fluorescence:
  PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~exc) -> PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~n)
  #deexcitation_Chld1_heat:
  PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~exc) -> PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~n)
  #excitation_transfer:
  PSII(P680~n,Qa~n,PhD1~n,ac~exc,ChlD1~n) <-> PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~exc)
  #primary_charge_separation_recombination:
  PSII(P680~n,Qa~n,PhD1~n,ac~n,ChlD1~exc) <-> PSII(P680~n,Qa~n,PhD1~m,ac~n,ChlD1~p)
  #stable_pair_generation_degeneration:
  PSII(P680~n,Qa~n,PhD1~m,ac~n,ChlD1~p) <-> PSII(P680~p,Qa~n,PhD1~m,ac~n,ChlD1~n)
  #quinone_qa_reduction_oxidation:
  PSII(P680~p,Qa~n,PhD1~m,ac~n,ChlD1~n) <-> PSII(P680~p,Qa~m,PhD1~n,ac~n,ChlD1~n)
end reaction rules
```

Many of the contexts in the rules could be removed. We therefore automatically apply a series of *specific unbound context eliminations* to get a reduced model. The result is the following:

```
begin reaction rules
  #deexcitation_antenna_fluorescence:
  PSII(ac~exc) -> PSII(ac~n)
  #deexcitation_antenna_heat:
  PSII(ac~exc) -> PSII(ac~n)
  #deexcitation_Chld1_fluorescence:
  PSII(ChlD1~exc) -> PSII(ChlD1~n)
  #deexcitation_Chld1_heat:
  PSII(ChlD1~exc) -> PSII(ChlD1~n)
  #excitation_transfer:
  PSII(ac~exc,ChlD1~n) <-> PSII(ac~n,ChlD1~exc)
  #primary_charge_separation_recombination:
  PSII(PhD1~n,ChlD1~exc) <-> PSII(PhD1~m,ChlD1~p)
  #stable_pair_generation_degeneration:
  PSII(P680~n,PhD1~m,ChlD1~p) <-> PSII(P680~p,PhD1~m,ChlD1~n)
  #quinone_qa_reduction_oxidation:
  PSII(Qa~n,PhD1~m,ChlD1~n) <-> PSII(Qa~m,PhD1~n,ChlD1~n)
end reaction rules
```

Specific unbound context elimination does not decrease the number of rules in the model. However, we can see that the rules become much more compact. The increased clarity of the rules makes it easy to see their effects.

In the *stable pair generation/degeneration* rule, one may notice the $\text{PhD1}\sim m$ specified in the context. Similarly, in the *quinone Qa reduction/oxidation* rule, $\text{ChlD1}\sim n$ is specified in the context. Next, we take a look on how the behavior of the model changes if we try to remove one of these contexts.

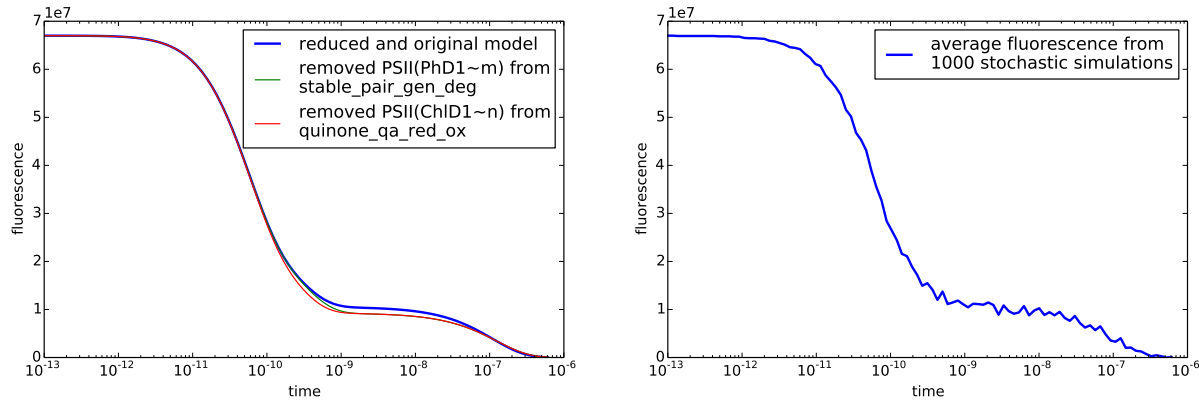


Figure 2: On the left: Fluorescence curve from the reduced model (blue) and the model where we manually remove also the context $\text{PhD1}\sim m$ (green) or $\text{ChlD1}\sim n$ (red). On the right: Fluorescence curve obtained by averaging 1000 stochastic simulations of the reduced model. The curves show fluorescence absorption time-course in relative units.

Figure 2 (left) compares fluorescence curves from several models. The blue line shows fluorescence for the original model. This curve is the same for the reduced model we get after automatically reducing the model. The green curve shows the changes that occur after we remove $\text{PhD1}\sim m$ from the *stable pair generation/degeneration* rule. We see that the fluorescence level is slightly lower on the plateau in the middle of the chart. Similar situation occurs when we remove $\text{ChlD1}\sim n$ from the context of *quinone Qa reduction/oxidation*. The resulting fluorescence levels are shown in red.

To illustrate why the fluorescence curves differ we compare the reaction networks of the reduced model and the reduced model with $\text{PhD1}\sim m$ removed from context. Figure 3 on the left shows the reaction network of the automatically reduced model. This reaction network is the same as the reaction network of the original model. On the right, the reaction network of the additionally reduced model is shown. We can see that this reaction network contains one more reachable agent that is not reachable in the original model. This difference results in the different fluorescence curves as shown before.

The reduced model have the same reaction network as the original model for the initial condition $\text{PSII}(\text{P680 } n, \text{Qa } n, \text{PhD1 } n, \text{ac } \text{exc}, \text{ChlD1 } n)$. For some other initial conditions, the original model would produce a reaction network consisting only of the initial solution, whereas the reduced model could produce a larger network. This should not matter, since the authors [16] make it explicit that the model should be used only with the specified initial condition.

4.4 Integrated Model of Light-reactions

Model Lazár [13] targets all important light-dependend thylakoid membrane processes participating in the *Z-scheme* of light. The model contains the following complexes and their parts: photosystem II (with parts P680, Qa, Qb); oxygen evolving complex (with states S_i where $i = 0, 1, 2, 3$); PQ, PQH - plastoquinone; cytochrome b_6f (with parts bL , bHc , f); Pc - plastocyanin; photosystem I (with parts

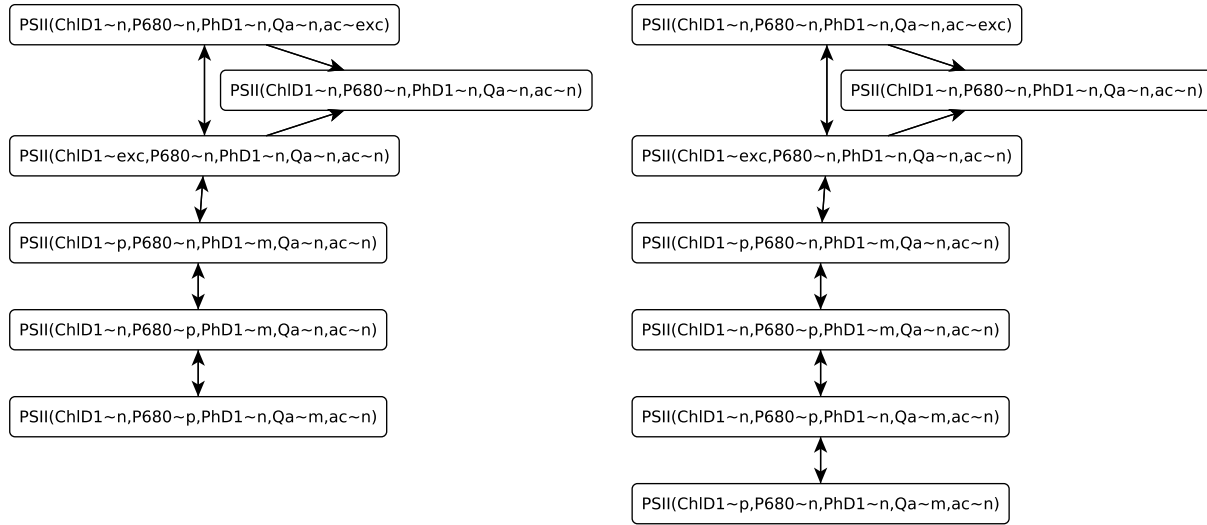


Figure 3: Reaction network of the reduced model (left) and of the reduced model with PhD1~m additionally removed from the context (right).

P700, Fb); Fd - ferredoxin; ferredoxin-NADP reductase. The model is expressed by 69 differential equations that can be browsed in detail at <http://www.e-photosynthesis.org>. As we show later, this model has great potential for succinct representation.

The rule-based reformulation of the full model in BNGL is the following:

```
begin molecule types
  PSII(P680~n~p, Qa~n~m, Qb~n~m~2m)
  PQ()
  PQH()
  S(x~0~1~2~3)
  CytB6F(bL~n, bHc~n, f~m)
  Fd(x~n~m)
  Pc(x~n~p)
  PSI(P700~n~p, Fb~n~m)
  FNR(x~i~a~am~a2m)
end molecule types

begin reaction rules
  #electron_transport_from_cytochrome_to_oxidized_Pc_1:
  CytB6F(bL~n, bHc~n, f~m) + Pc(x~p) <-> CytB6F(bL~n, bHc~n, f~n) + Pc(x~n)
  #electron_transport_from_cytochrome_to_oxidized_Pc_2:
  CytB6F(bL~m, bHc~n, f~m) + Pc(x~p) <-> CytB6F(bL~m, bHc~n, f~n) + Pc(x~n)
  #electron_transport_from_cytochrome_to_oxidized_Pc_3:
  CytB6F(bL~n, bHc~m, f~m) + Pc(x~p) <-> CytB6F(bL~n, bHc~m, f~n) + Pc(x~n)
  #electron_transport_from_cytochrome_to_oxidized_Pc_4:
  CytB6F(bL~m, bHc~m, f~m) + Pc(x~p) <-> CytB6F(bL~m, bHc~m, f~n) + Pc(x~n)
  #electron_transport_from_cytochrome_to_oxidized_Pc_5:
  CytB6F(bL~n, bHc~2m, f~m) + Pc(x~p) <-> CytB6F(bL~n, bHc~2m, f~n) + Pc(x~n)
  #electron_transport_from_cytochrome_to_oxidized_Pc_6:
```

```

CytB6F(bL~m,bHc~2m,f~m) + Pc(x~p) <-> CytB6F(bL~m,bHc~2m,f~n) + Pc(x~n)
#electron_transport_from_reduced_Fd_to_cytochrome_1:
Fd(x~m) + CytB6F(bL~n,bHc~n,f~n) <-> Fd(x~n) + CytB6F(bL~n,bHc~m,f~n)
#electron_transport_from_reduced_Fd_to_cytochrome_2:
Fd(x~m) + CytB6F(bL~m,bHc~n,f~n) <-> Fd(x~n) + CytB6F(bL~m,bHc~m,f~n)
#electron_transport_from_reduced_Fd_to_cytochrome_3:
Fd(x~m) + CytB6F(bL~n,bHc~n,f~m) <-> Fd(x~n) + CytB6F(bL~n,bHc~m,f~m)
#electron_transport_from_reduced_Fd_to_cytochrome_4:
Fd(x~m) + CytB6F(bL~m,bHc~n,f~m) <-> Fd(x~n) + CytB6F(bL~m,bHc~m,f~m)
#electron_transport_from_reduced_Fd_to_cytochrome_5:
Fd(x~m) + CytB6F(bL~n,bHc~m,f~n) <-> Fd(x~n) + CytB6F(bL~n,bHc~2m,f~n)
#electron_transport_from_reduced_Fd_to_cytochrome_6:
Fd(x~m) + CytB6F(bL~m,bHc~m,f~n) <-> Fd(x~n) + CytB6F(bL~m,bHc~2m,f~n)
#electron_transport_from_reduced_Fd_to_cytochrome_7:
Fd(x~m) + CytB6F(bL~n,bHc~m,f~m) <-> Fd(x~n) + CytB6F(bL~n,bHc~2m,f~m)
#electron_transport_from_reduced_Fd_to_cytochrome_8:
Fd(x~m) + CytB6F(bL~m,bHc~m,f~m) <-> Fd(x~n) + CytB6F(bL~m,bHc~2m,f~m)
#electron_transport_inside_cytochrome_1:
CytB6F(bL~m,bHc~n,f~n) <-> CytB6F(bL~n,bHc~m,f~n)
#electron_transport_inside_cytochrome_2:
CytB6F(bL~m,bHc~n,f~m) <-> CytB6F(bL~n,bHc~m,f~m)
#electron_transport_inside_cytochrome_3:
CytB6F(bL~m,bHc~m,f~n) <-> CytB6F(bL~n,bHc~2m,f~n)
#electron_transport_inside_cytochrome_4:
CytB6F(bL~m,bHc~m,f~m) <-> CytB6F(bL~n,bHc~2m,f~m)
#electron_transport_from_reduced_PQ_to_cytochrome_1:
PQH() + CytB6F(bL~n,bHc~n,f~n) <-> PQ() + CytB6F(bL~m,bHc~n,f~m)
#electron_transport_from_reduced_PQ_to_cytochrome_2:
PQH() + CytB6F(bL~n,bHc~m,f~n) <-> PQ() + CytB6F(bL~m,bHc~m,f~m)
#electron_transport_from_reduced_PQ_to_cytochrome_3:
PQH() + CytB6F(bL~n,bHc~2m,f~n) <-> PQ() + CytB6F(bL~m,bHc~2m,f~m)
#electron_transport_from_cytochrome_to_oxidized_PQ_1:
PQ() + CytB6F(bL~n,bHc~2m,f~n) <-> PQH() + CytB6F(bL~n,bHc~n,f~n)
#electron_transport_from_cytochrome_to_oxidized_PQ_2:
PQ() + CytB6F(bL~m,bHc~2m,f~n) <-> PQH() + CytB6F(bL~m,bHc~n,f~n)
#electron_transport_from_cytochrome_to_oxidized_PQ_4:
PQ() + CytB6F(bL~m,bHc~2m,f~m) <-> PQH() + CytB6F(bL~m,bHc~n,f~m)
#electron_transport_from_cytochrome_to_oxidized_PQ_3:
PQ() + CytB6F(bL~n,bHc~2m,f~m) <-> PQH() + CytB6F(bL~n,bHc~n,f~m)
#electron_transport_from_Fd_to_FNR_1:
FNR(x~a) + Fd(x~m) <-> FNR(x~am) + Fd(x~n)
#electron_transport_from_Fd_to_FNR_2:
FNR(x~am) + Fd(x~m) <-> FNR(x~a2m) + Fd(x~n)
#electron_transport_from_photosystemI_to_Fd_1:
Fd(x~n) + PSI(P700~n,Fb~m) <-> Fd(x~m) + PSI(P700~n,Fb~n)
#electron_transport_from_photosystemI_to_Fd_2:
Fd(x~n) + PSI(P700~p,Fb~m) <-> Fd(x~m) + PSI(P700~p,Fb~n)
#activation_of_FNR:
FNR(x~i) -> FNR(x~a)
#turnover_of_FNR:
FNR(x~a2m) -> FNR(x~a)
#electron_donation_1_1:

```

```

S(x~0) + PSII(P680~p,Qa~n,Qb~n) -> S(x~1) + PSII(P680~n,Qa~n,Qb~n)
#electron_donation_2_1:
S(x~0) + PSII(P680~p,Qa~m,Qb~n) -> S(x~1) + PSII(P680~n,Qa~m,Qb~n)
#electron_donation_3_1:
S(x~0) + PSII(P680~p,Qa~n,Qb~m) -> S(x~1) + PSII(P680~n,Qa~n,Qb~m)
#electron_donation_4_1:
S(x~0) + PSII(P680~p,Qa~m,Qb~m) -> S(x~1) + PSII(P680~n,Qa~m,Qb~m)
#electron_donation_5_1:
S(x~0) + PSII(P680~p,Qa~n,Qb~2m) -> S(x~1) + PSII(P680~n,Qa~n,Qb~2m)
#electron_donation_6_1:
S(x~0) + PSII(P680~p,Qa~m,Qb~2m) -> S(x~1) + PSII(P680~n,Qa~m,Qb~2m)
#electron_donation_1_2:
S(x~1) + PSII(P680~p,Qa~n,Qb~n) -> S(x~2) + PSII(P680~n,Qa~n,Qb~n)
#electron_donation_2_2:
S(x~1) + PSII(P680~p,Qa~m,Qb~n) -> S(x~2) + PSII(P680~n,Qa~m,Qb~n)
#electron_donation_3_2:
S(x~1) + PSII(P680~p,Qa~n,Qb~m) -> S(x~2) + PSII(P680~n,Qa~n,Qb~m)
#electron_donation_4_2:
S(x~1) + PSII(P680~p,Qa~m,Qb~m) -> S(x~2) + PSII(P680~n,Qa~m,Qb~m)
#electron_donation_5_2:
S(x~1) + PSII(P680~p,Qa~n,Qb~2m) -> S(x~2) + PSII(P680~n,Qa~n,Qb~2m)
#electron_donation_6_2:
S(x~1) + PSII(P680~p,Qa~m,Qb~2m) -> S(x~2) + PSII(P680~n,Qa~m,Qb~2m)
#electron_donation_1_3:
S(x~2) + PSII(P680~p,Qa~n,Qb~n) -> S(x~3) + PSII(P680~n,Qa~n,Qb~n)
#electron_donation_2_3:
S(x~2) + PSII(P680~p,Qa~m,Qb~n) -> S(x~3) + PSII(P680~n,Qa~m,Qb~n)
#electron_donation_3_3:
S(x~2) + PSII(P680~p,Qa~n,Qb~m) -> S(x~3) + PSII(P680~n,Qa~n,Qb~m)
#electron_donation_4_3:
S(x~2) + PSII(P680~p,Qa~m,Qb~m) -> S(x~3) + PSII(P680~n,Qa~m,Qb~m)
#electron_donation_5_3:
S(x~2) + PSII(P680~p,Qa~n,Qb~2m) -> S(x~3) + PSII(P680~n,Qa~n,Qb~2m)
#electron_donation_6_3:
S(x~2) + PSII(P680~p,Qa~m,Qb~2m) -> S(x~3) + PSII(P680~n,Qa~m,Qb~2m)
#electron_donation_1_0:
S(x~3) + PSII(P680~p,Qa~n,Qb~n) -> S(x~0) + PSII(P680~n,Qa~n,Qb~n)
#electron_donation_2_0:
S(x~3) + PSII(P680~p,Qa~m,Qb~n) -> S(x~0) + PSII(P680~n,Qa~m,Qb~n)
#electron_donation_3_0:
S(x~3) + PSII(P680~p,Qa~n,Qb~m) -> S(x~0) + PSII(P680~n,Qa~n,Qb~m)
#electron_donation_4_0:
S(x~3) + PSII(P680~p,Qa~m,Qb~m) -> S(x~0) + PSII(P680~n,Qa~m,Qb~m)
#electron_donation_5_0:
S(x~3) + PSII(P680~p,Qa~n,Qb~2m) -> S(x~0) + PSII(P680~n,Qa~n,Qb~2m)
#electron_donation_6_0:
S(x~3) + PSII(P680~p,Qa~m,Qb~2m) -> S(x~0) + PSII(P680~n,Qa~m,Qb~2m)
#electron_transports_from_qa_to_qb_1:
PSII(P680~p,Qa~m,Qb~n) <-> PSII(P680~p,Qa~n,Qb~m)
#electron_transports_from_qa_to_qb_2:
PSII(P680~n,Qa~m,Qb~n) <-> PSII(P680~n,Qa~n,Qb~m)
#electron_transports_from_qa_to_reduced_qb_1:

```

```

PSII(P680~p,Qa~m,Qb~m) <-> PSII(P680~p,Qa~n,Qb~2m)
#electron_transports_from_qa_to_reduced_qb_2:
PSII(P680~n,Qa~m,Qb~m) <-> PSII(P680~n,Qa~n,Qb~2m)
#Qb_PQ_exchange_1:
PQ() + PSII(P680~p,Qa~n,Qb~2m) <-> PQH() + PSII(P680~p,Qa~n,Qb~n)
#Qb_PQ_exchange_2:
PQ() + PSII(P680~n,Qa~n,Qb~2m) <-> PQH() + PSII(P680~n,Qa~n,Qb~n)
#Qb_PQ_exchange_3:
PQ() + PSII(P680~p,Qa~m,Qb~2m) <-> PQH() + PSII(P680~p,Qa~m,Qb~n)
#Qb_PQ_exchange_4:
PQ() + PSII(P680~n,Qa~m,Qb~2m) <-> PQH() + PSII(P680~n,Qa~m,Qb~n)
#electron_donation_to_P700_by_Pc_1:
Pc(x~n) + PSI(P700~p,Fb~n) <-> Pc(x~p) + PSI(P700~n,Fb~n)
#electron_donation_to_P700_by_Pc_2:
Pc(x~n) + PSI(P700~p,Fb~m) <-> Pc(x~p) + PSI(P700~n,Fb~m)
#charge_separation_in_PSI:
PSI(P700~n,Fb~n) -> PSI(P700~p,Fb~m)
#charge_separation_in_PSII_1:
PSII(P680~n,Qa~n,Qb~n) -> PSII(P680~p,Qa~m,Qb~n)
#charge_separation_in_PSII_2:
PSII(P680~n,Qa~n,Qb~m) -> PSII(P680~p,Qa~m,Qb~m)
#charge_separation_in_PSII_3:
PSII(P680~n,Qa~n,Qb~2m) -> PSII(P680~p,Qa~m,Qb~2m)
end reaction rules

```

The model has many rules that are perfect candidates for reduction using *context enumeration elimination*, such as the following three *charge separation* rules:

```

PSII(P680~n,Qa~n,Qb~n) -> PSII(P680~p,Qa~m,Qb~n)
PSII(P680~n,Qa~n,Qb~m) -> PSII(P680~p,Qa~m,Qb~m)
PSII(P680~n,Qa~n,Qb~2m) -> PSII(P680~p,Qa~m,Qb~2m)

```

These rules can be reduced to the following single *charge separation* rule:

```

PSII(P680~n,Qa~n,Qb) -> PSII(P680~p,Qa~m,Qb)

```

After applying all *context enumeration eliminations* we are able to reduce the model size from 69 rules down to just 22 rules. Since the model is not using any binding sites, we can automatically apply *generic unbound context eliminations* to further reduce the model. The rule stated above is reduced to the following form:

```

PSII(P680~n,Qa~n) -> PSII(P680~p,Qa~m)

```

We managed to reduce the model significantly. The original ideas of the author are much more obvious in this reduced model. The reduced model is much easier to modify and extend than its original version. The resulting model is the following:

```

begin reaction rules
#electron_transport_from_Fd_to_FNR:
  FNR(x~a) + Fd(x~m) <-> FNR(x~am) + Fd(x~n)
  FNR(x~am) + Fd(x~m) <-> FNR(x~a2m) + Fd(x~n)
#activation_of_FNR:
  FNR(x~i) -> FNR(x~a)
#turnover_of_FNR:

```

```

FNR(x~a2m) -> FNR(x~a)
#charge_separation_in_PSI:
PSI(P700~n,Fb~n) -> PSI(P700~p,Fb~m)
#electron_transport_from_photosystemI_to_Fd:
Fd(x~n) + PSI(Fb~m) <-> Fd(x~m) + PSI(Fb~n)
#electron_transports_from_qa_to_qb:
PSII(Qa~m,Qb~n) <-> PSII(Qa~n,Qb~m)
#electron_transports_from_qa_to_reduced_qb:
PSII(Qa~m,Qb~m) <-> PSII(Qa~n,Qb~2m)
#electron_transport_inside_cytochrome:
CytB6F(bL~m,bHc~n) <-> CytB6F(bL~n,bHc~m)
CytB6F(bL~m,bHc~m) <-> CytB6F(bL~n,bHc~2m)
#electron_transport_from_reduced_PQ_to_cytochrome:
PQH() + CytB6F(bL~n,f~n) <-> PQH() + CytB6F(bL~m,f~m)
#charge_separation_in_PSII:
PSII(P680~n,Qa~n) -> PSII(P680~p,Qa~m)
#electron_donation_to_P700_by_Pc:
Pc(x~n) + PSI(P700~p) <-> Pc(x~p) + PSI(P700~n)
#electron_transport_from_cytochrome_to_oxidized_PQ:
PQH() + CytB6F(bHc~2m) <-> PQH() + CytB6F(bHc~n)
#electron_transport_from_reduced_Fd_to_cytochrome:
Fd(x~m) + CytB6F(bHc~n) <-> Fd(x~n) + CytB6F(bHc~m)
Fd(x~m) + CytB6F(bHc~m) <-> Fd(x~n) + CytB6F(bHc~2m)
#electron_donation:
S(x~0) + PSII(P680~p) -> S(x~1) + PSII(P680~n)
S(x~1) + PSII(P680~p) -> S(x~2) + PSII(P680~n)
S(x~2) + PSII(P680~p) -> S(x~3) + PSII(P680~n)
S(x~3) + PSII(P680~p) -> S(x~0) + PSII(P680~n)
#Qb_PQ_exchange:
PQH() + PSII(Qb~2m) <-> PQH() + PSII(Qb~n)
#electron_transport_from_cytochrome_to_oxidized_Pc:
CytB6F(f~m) + Pc(x~p) <-> CytB6F(f~n) + Pc(x~n)
end reaction rules

```

For the purpose of further analysis, the following observables computing the fluorescence emission are defined [13]:

```

begin observables
Molecules Qa_m      PSII(Qa~m)
Molecules PQ_n      PQH()
end observables

```

The computed fluorescence signals are defined by the following assignments over the observables:

- *unquenched fluorescence*:

$$F_{unq}(t) = \frac{0.45Qa_m(t)}{1 - 0.55Qa_m(t)}$$

- *quenched fluorescence*:

$$F_q(t) = \frac{F_{unq}(t)}{1 + (\frac{1}{45} + \frac{4}{63}Qa_m(t))PQ_n(t)}$$

Figure 4 shows signals obtained by averaging the results from 20 stochastic simulations of the model. We had to convert the deterministic rates to stochastic ones to obtain these results. Details of this conversion can be seen in [17].

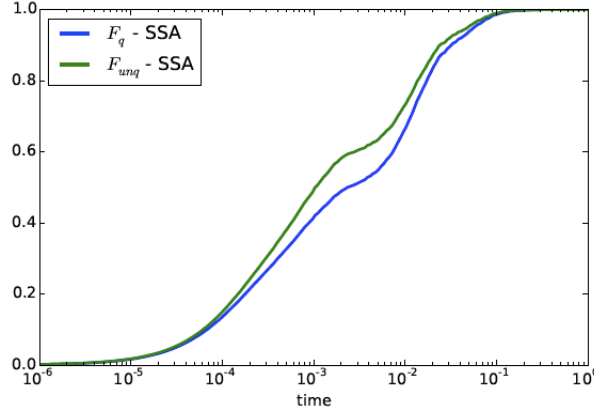


Figure 4: Averaged signals computed from the outputs of 20 stochastic simulations of Lazár's model. The curves show the time course of light emission intensity in relative units.

In the model presented above, a module of photosystem II including the oxygen evolving complex is considered. In [14], Lazár elaborates on several models of photosystem II with varying levels of complexity integrating the following components of PSII:

- Electron donor $P680$
- S_i states of the oxygen evolving complex (OEC) for $i = 0, 1, 2, 3$.
- Primary quinone electron acceptor Q_a
- Secondary quinone electron acceptor Q_b
- PQ (plastoquinone) pool

The measured output of the models is demonstrated in Figure 5 – the fluorescence rise (FLR) in chlorophyll a .

Lazár considers several simplifications of the PSII module and describes their effect on the shape of the fluorescence signal:

1. The S_i -states of the OEC are considered as separated from PSII. In BNGL (specifying only the relevant sites), instead of PSII(S₀~1~2~3) we would have PSII() and S(p₀~1~2~3). When describing the model using differential equations this simplification reduces the number of equations from 64 down to 20. With this simplification the system turns into second order kinetics instead of first order kinetics. Illustrated in BNGL:
 $\text{PSII}(\text{P680}_{\sim p}, \text{S}_{\sim 0}) \rightarrow \text{PSII}(\text{P680}_{\sim n}, \text{S}_{\sim 1})$ changes to
 $\text{PSII}(\text{P680}_{\sim p}) + \text{S}(\text{p}_{\sim 0}) \rightarrow \text{PSII}(\text{P680}_{\sim n}) + \text{S}(\text{p}_{\sim 1})$
2. Exchange of the double reduced Q_b with a PQ molecule from the PQ pool is described by one second order reaction instead of two subsequent reactions. In BNGL, instead of
 $\text{PSII}(\text{Qb}_{!1}) . \text{PQ}(\text{Qb}_{\sim 2m!1}) \leftrightarrow \text{PSII}(\text{Qb}) + \text{PQ}(\text{Qb}_{\sim 2m})$ and
 $\text{PSII}(\text{Qb}) + \text{PQ}(\text{Qb}_{\sim n}) \leftrightarrow \text{PSII}(\text{Qb}_{!1}) . \text{PQ}(\text{Qb}_{\sim n!1})$
 we have
 $\text{PSII}(\text{Qb}_{\sim 2m}) + \text{PQ}(\text{Qb}_{\sim n}) \leftrightarrow \text{PSII}(\text{Qb}_{\sim n}) + \text{PQ}(\text{Qb}_{\sim 2m})$.
 This simplification makes the reading of the model a bit easier. By using this simplification along with the previous one the model is further reduced to 16 equations.

3. Only equations that follow a logical order are taken into account.

These simplifications are combined into 8 models (two unique choices for each simplification – as shown in Table 1). Different model formulations lead to different simulations of the fluorescence rise. Lazár concludes that it is very important to care about how we formulate the model, because some simplifications may lead to completely unusual FLR behaviour. Models 7 and 8 resulted in a behaviour very similar to the O-J-I-P behaviour observed in *in vivo* experiments. We know that OEC is actually bound to PSII and that the process of reducing PQ by Q_b proceeds in two steps. These are two more arguments that proof the model 8 to be the most accurate representation of the real-world system. As we present later, the size of model 8 is greatly reduced by rule-based modelling. Its size is even smaller than the size of the smallest model listed here.

| Model no.: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---------------------------|-----|-----|-----|-----|-----|-----|-----|----|
| OEC separated from PSII | yes | yes | yes | yes | no | no | no | no |
| PQ exchange by 1 reaction | yes | yes | no | no | yes | yes | no | no |
| Only logical equations | no | yes | no | yes | yes | no | yes | no |

Table 1: Model variants of PSII considered by Lazár.

To demonstrate the advantage of rule-based modelling to easily modify and compose the individual model variants, we take the most complex model of PSII Lazár presented in [14] (model 8) and we reformulate it in BNGL. Simulation of this model is consistent with O-J-I-P transient (see Figure 5). In its original form, this model consists of 64 differential equations that can be rewritten to following 10 rules:

```
begin reaction rules
  # Light induced charge separation between P680 and Qa and charge recombination:
  PSII(P680~n,Qa~n) <-> PSII(P680~p,Qa~m)
  # Electron donation from S-states of OEC to P680+
  PSII(P680~p,S~0) -> PSII(P680~n,S~1)
  PSII(P680~p,S~1) -> PSII(P680~n,S~2)
  PSII(P680~p,S~2) -> PSII(P680~n,S~3)
  PSII(P680~p,S~3) -> PSII(P680~n,S~0)
  # Electron transport from Qa- to Qb:
  PSII(Qa~m,Qb~red!1).PQ(Qb~n!1) <-> PSII(Qa~n,Qb~red!1).PQ(Qb~m!1)
  PSII(Qa~m,Qb~red!1).PQ(Qb~m!1) <-> PSII(Qa~n,Qb~red!1).PQ(Qb~2m!1)
  # Exchange of doubly reduced Qb with oxidized PQ molecule
  # from the PQ pool by two subsequent reversible reactions:
  PSII(Qb~red!1).PQ(Qb~2m!1) <->
    PSII(Qb~red) + PQ(Qb~2m)
  PSII(Qb~red) + PQ(Qb~n) <->
    PSII(Qb~red!1).PQ(Qb~n!1)
  # Reversible reoxidation of reduced PQ molecules from PQ pool:
  PQ(Qb~2m) <-> PQ(Qb~n)
end reaction rules
```

Next, we replace the original photosystem II module in the integrated model with model 8. This modification would be very hardly realisable when using the ODE approach to express the models. The resulting model is listed below.

```
begin molecule types
```

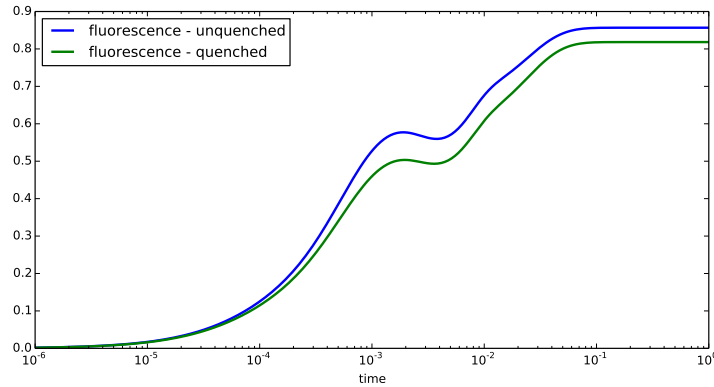


Figure 5: The fluorescence curves from the model of photosystem II. The curves show the time course of light emission intensity in relative units.

```

PSII(P680~n~p,Qa~n~m,S~0~1~2~3,Qb~n~m)
PQ(Qb~n~m~2m)
CytB6F(bL~n~m,bHc~n~m~2m,f~n~m)
Fd(x~n~m)
Pc(x~n~p)
PSI(P700~n~p,Fb~n~m)
FNR(x~i~a~am~a2m)
end molecule types

begin reaction rules
# Light induced charge separation/recombination between P680 and Qa:
PSII(P680~n,Qa~n) <-> PSII(P680~p,Qa~m)
# Electron donation from S-states of OEC to P680+
PSII(P680~p,S~0) -> PSII(P680~n,S~1)
PSII(P680~p,S~1) -> PSII(P680~n,S~2)
PSII(P680~p,S~2) -> PSII(P680~n,S~3)
PSII(P680~p,S~3) -> PSII(P680~n,S~0)
# Electron transport from Qa- to Qb:
PSII(Qa~m,Qb~m!1).PQ(Qb~n!1) <-> PSII(Qa~n,Qb~m!1).PQ(Qb~m!1)
PSII(Qa~m,Qb~m!1).PQ(Qb~m!1) <-> PSII(Qa~n,Qb~m!1).PQ(Qb~2m!1)
# Exchange of doubly reduced Qb with oxidized PQ molecule
# from the PQ pool by two subsequent reversible reactions:
PSII(Qb~m!1).PQ(Qb~2m!1) <-> PSII(Qb~m) + PQ(Qb~2m)
PSII(Qb~m) + PQ(Qb~n) <-> PSII(Qb~m!1).PQ(Qb~n!1)
# Electron transport from Fd to FNR:
FNR(x~a) + Fd(x~m) <-> FNR(x~am) + Fd(x~n)
FNR(x~am) + Fd(x~m) <-> FNR(x~a2m) + Fd(x~n)
# Activation of FNR:
FNR(x~i) -> FNR(x~a)
# Turnover of FNR:
FNR(x~a2m) -> FNR(x~a)
# Charge separation in PSI:
PSI(P700~n,Fb~n) -> PSI(P700~p,Fb~m)

```

```

# Electron transport from PSI to Fd:
Fd(x~n) + PSI(Fb~m) <-> Fd(x~m) + PSI(Fb~n)
# Electron transport inside cytochrome:
CytB6F(bL~m,bHc~m) <-> CytB6F(bL~n,bHc~2m)
CytB6F(bL~m,bHc~n) <-> CytB6F(bL~n,bHc~m)
# Electron transport from reduced PQ to cytochrome:
PQ(Qb~2m) + CytB6F(bL~n,f~n) <-> PQ(Qb~n) + CytB6F(bL~m,f~m)
# Electron donation to P700 by Pc:
Pc(x~n) + PSI(P700~p) <-> Pc(x~p) + PSI(P700~n)
# Electron transport from cytochrome to oxidized PQ:
PQ(Qb~n) + CytB6F(bHc~2m) <-> PQ(Qb~2m) + CytB6F(bHc~n)
# Electron transport from reduced Fd to cytochrome:
Fd(x~m) + CytB6F(bHc~n) <-> Fd(x~n) + CytB6F(bHc~m)
Fd(x~m) + CytB6F(bHc~m) <-> Fd(x~n) + CytB6F(bHc~2m)
# Electron transport from cytochrome to oxidized Pc:
CytB6F(f~m) + Pc(x~p) <-> CytB6F(f~n) + Pc(x~n)
end reaction rules

```

In Figure 6, the outputs of the original and the combined model are compared.

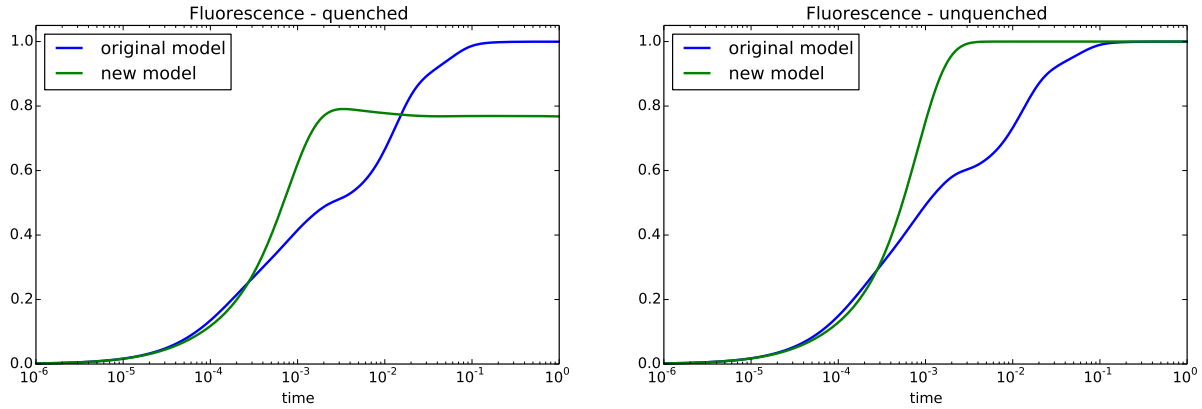


Figure 6: Comparison of the fluorescence curves of the original and the combined model.

5 Conclusions

We have demonstrated the unsuitability of traditional reaction-based modelling approaches for modelling complex biochemical processes, such as photosynthesis. We explored existing models of photosynthesis and described the simplifications that were made in those models in order to battle the problem of combinatorial explosion. We showed how these simplifications are undesirable. Rule-based modelling allows us to compactly model the processes of photosynthesis in their full mechanistic complexity without the need for such simplifying assumptions.

We set on to naively reformulate selected representative models of photosynthesis as rule-based models. These reformulated models were unnecessarily large, not exploiting the advantages of the rule-based format. Therefore, we formally defined several intuitive syntactic operations that can be used to reduce the size of these models. We provided a case study where we implemented these operations

so they can be performed automatically and we managed to achieve large reductions in the size of the models. The order in which we applied the reductions turned out to be satisfactory.

We believe that in the future, the communities of biologists who are modelling photosynthesis consider the use of rule-based modelling. Rule-based modelling brings in many advantages and eliminates the reason of some artificial model simplifications.

References

- [1] Liam Brennan & Philip Owende (2010): *Biofuels from microalgae A review of technologies for production, processing, and extractions of biofuels and co-products*. *Renewable and Sustainable Energy Reviews* 14(2), pp. 557 – 577.
- [2] Ferdinanda Camporesi, Jérôme Feret, Heinz Koepl & Tatjana Petrov (2010): *Combining model reductions*. *Electronic Notes in Theoretical Computer Science* 265, pp. 73–96.
- [3] Lily A Chylek, Leonard A Harris, Chang-Shung Tung, James R Faeder, Carlos F Lopez & William S Hlavacek (2014): *Rule-based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems*. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 6(1), pp. 13–36.
- [4] Lily A. Chylek, Edward C. Stites, Richard G. Posner & William S. Hlavacek (2013): *Innovations of the Rule-Based Modeling Approach*. In: Ale Prokop & Bla Csuks, editors: *Systems Biology*. Springer Netherlands, pp. 273–300.
- [5] Vincent Danos (2009): *Agile Modelling of Cellular Signalling (Invited Paper)*. *Electron. Notes Theor. Comput. Sci.* 229(4), pp. 3–10.
- [6] Vincent Danos, Jrme Feret, Walter Fontana & Jean Krivine (2008): *Abstract Interpretation of Cellular Signalling Networks*. In: Francesco Logozzo, DoronA. Peled & LenoreD. Zuck, editors: *Verification, Model Checking, and Abstract Interpretation, Lecture Notes in Computer Science* 4905. Springer Berlin Heidelberg, pp. 83–97.
- [7] Vincent Danos & Cosimo Laneve (2004): *Formal molecular biology*. *Theoretical Computer Science* 325(1), pp. 69–110.
- [8] William S Hlavacek, James R Faeder, Michael L Blinov, Richard G Posner, Michael Hucka & Walter Fontana (2006): *Rules for modeling signal-transduction systems*. *Science Signaling* 2006(344), p. re6.
- [9] AR Holzwarth, MG Müller, M Reus, M Nowaczyk, J Sander & M Rögner (2006): *Kinetics and mechanism of electron transfer in intact photosystem II and in the isolated reaction center: pheophytin is the primary electron acceptor*. *Proceedings of the National Academy of Sciences* 103(18), pp. 6895–6900.
- [10] Christer Jansson & Trent Northen (2010): *Calcifying cyanobacteria the potential of biomineralization for carbon capture and storage*. *Current Opinion in Biotechnology* 21(3), pp. 365 – 371. *Energy biotechnology Environmental biotechnology*.
- [11] Heinz Koepl, Marc Hafner & Vincent Danos (2009): *Rule-based modeling for protein-protein interaction networks - the cyanobacterial circadian clock as a case study*. In: *International Workshop on Computational Systems Biology 2009*.
- [12] Agu Laisk, Ladislav Nedbal & Govindjee (2009): *Photosynthesis in silico*. Springer.
- [13] D. Lazár (2009): *Modelling of light-induced chlorophyll a fluorescence rise (O-J-I-P transient) and changes in 820 nm-transmittance signal of photosynthesis*. *Photosynthetica* 47(4), pp. 483–498.
- [14] Dušan Lazár & Jiří Jablonský (2009): *On the approaches applied in formulation of a kinetic model of photosystem II: Different approaches lead to different simulations of the chlorophyll a fluorescence transients*. *Journal of theoretical biology* 257(2), pp. 260–269.
- [15] Carlos F Lopez, Jeremy L Muhlich, John A Bachman & Peter K Sorger (2013): *Programming biological models in Python using PySB*. *Molecular systems biology* 9(1).

- [16] Ladislav Nedbal, Jan Červený, Uwe Rascher & Henning Schmidt (2007): *E-photosynthesis: a comprehensive modeling approach to understand chlorophyll fluorescence transients and other complex dynamic features of photosynthesis in fluctuating light*. *Photosynthesis research* 93(1-3), pp. 223–234.
- [17] Juraj Nižnan (2014). *Compact representation of biochemical processes in photosynthesis*. Master's Thesis, Masaryk University.
- [18] Valentin I Prokhorenko & Alfred R Holzwarth (2000): *Primary processes and structure of the photosystem II reaction center: a photon echo study*. *The Journal of Physical Chemistry B* 104(48), pp. 11563–11578.
- [19] Uwe Rascher & Ladislav Nedbal (2006): *Dynamics of photosynthesis in fluctuating light*. *Current opinion in plant biology* 9(6), pp. 671–678.
- [20] Adam Smith, Wen Xu, Yao Sun, James Faeder & G Elisabeta Marai (2012): *RuleBender: integrated modeling, simulation and visualization for rule-based intracellular biochemistry*. *BMC Bioinformatics* 13(Suppl 8), p. S3.
- [21] David Šafránek, Jan Červený, Matěj Klement, Jana Pospíšilová, Luboš Brim, Dušan Lazár & Ladislav Nedbal (2011): *E-photosynthesis: Web-based platform for modeling of complex photosynthetic processes*. *BioSystems* 103(2), pp. 115–124.
- [22] Chang-Peng Xin, Jin Yang & Xin-Guang Zhu (2013): *A model of chlorophyll a fluorescence induction kinetics with explicit description of structural constraints of individual photosystem II units*. *Photosynthesis Research* 117(1-3), pp. 339–354.